



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/564,106	01/09/2006	Alex Zakonov	AVI-0003	7200
7590 12/17/2009 Law Offices of Steven McHugh 46 Washington Street Middletown, CT 06457				
EXAMINER				
PATEL, JIGAR P				
ART UNIT		PAPER NUMBER		
2114				
MAIL DATE		DELIVERY MODE		
12/17/2009		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

DETAILED ACTION

1. This action is in response to the filing on September 14, 2009. Claims 1-4, 7-16, 18, and 19 are pending and have been considered below. The applicant has canceled claims 5, 6, 17, and 20.

Priority

2. Acknowledgement is made of applicant's priority of United States Provisional Patent Application Number 60/486,560 filed on July 11, 2003.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-4, 7-16, 18, and 19 are rejected under 35 U.S.C. 102(e) as being anticipated by Ritz et al. (US 7,263,632 B2).

As per claim 1: **A method for monitoring application exceptions generated by a software application, comprising:**
operating the software application to generate exception event data responsive to an

application exception;

dynamically identifying said application exception generated by the software application prior to said application exception being logged

Ritz discloses [col. 06, lines 38-50] the monitoring service applies policy to filter which events are propagated up to invoke diagnostic module for root cause determination. Examples of such events may be desirable include enterprise environments where an IT manager or system administrator may prefer that the operating system not perform certain automatic root cause determination and/or problem resolutions actions automatically. Therefore, the operating system is capable of dynamically identifying root cause determination.

obtaining exception data responsive to said application exception;

examining said exception data, prior to said application exception being logged, to determine whether said application exception is a critical exception and to identify critical exception data;

determining type of said critical exception; and

Ritz discloses [col. 05, lines 27-43] the event provider communicate events to a logger. Accordingly, any given event provider need not generate an event for every interaction it senses, but may generate only the more relevant events relating to root causes of problems [identifying critical exception]. For example, an event need not be generated every time a disk drive writes to a sector [non-critical event]. However, an event might be generated if the disk drive fails to respond to a read or write command [critical event]. The logger makes these determinations before the event has been logged to the event trace log file.

processing said critical exception data responsive to said type of said critical exception event.

Ritz further discloses [Abstract] the diagnostic module queries the log file to correlate events relevant to diagnosis of the problem, and identifies the root cause by evaluating the results of the query. Once the root cause of the problem is diagnosed, a resolution module corresponding to that root cause may be invoked to programmatically resolve the problem.

As per claim 2: **The method of Claim 1, wherein said operating includes operating said software application in at least one of a .NET framework and a J2EE framework [Abstract; operating system (Windows)].**

As per claim 3: **The method of Claim 1, wherein said determining includes determining whether said type of said critical exception is at least one of a primary critical exception event and a derived critical exception.**

Ritz discloses [Abstract] events are monitored within an operating system, and at least a subset of the events are logged to a log file. It is understood that a new event logged would be a primary critical exception. If a similar log exists, it would be a derived critical exception event.

As per claim 4: **The method of Claim 3, wherein said processing includes processing said critical exception data responsive to at least one of said primary critical exception and said derived critical exception.**

Ritz discloses [Abstract] events are monitored within an operating system, and at least a subset of the events are logged to a log file. It is understood that a new event logged would be a primary critical exception. If a similar log exists, it would be a derived critical exception event.

As per claim 7: **The method of Claim 1, wherein said examining includes examining said critical exception data to determine if an exception chain exists.**

Ritz discloses [col. 06, lines 14-22] the diagnostics policy service determines when an actual problem has occurred by, for example, detecting a predetermined single error condition, or by detecting a predetermined sequence of error conditions has arisen [exception chain exists].

As per claim 8: **The method of Claim 7, wherein said processing further includes collecting critical exception data responsive to said critical exception and creating an exception information database.**

Ritz discloses [Abstract] events are monitored within an operating system, and at least a subset of the events are logged to a log file [information database].

As per claim 9: **The method of Claim 8, wherein said processing further includes creating said critical exception chain and associating said collected critical exception data with said critical exception chain.**

Ritz discloses [col. 06, lines 14-22] the diagnostics policy service determines when an actual problem has occurred by, for example, detecting a predetermined single error condition, or by detecting a predetermined sequence of error conditions has arisen [exception chain exists]. Ritz

further discloses [col. 06, lines 23-29] once a problem is detected, the computing system performs a functional result-oriented step for programmatically diagnosing a problem evidenced by the one or more error conditions.

As per claim 10: **The method of Claim 7, wherein said processing includes associating said collected critical exception data with said critical exception chain.**

Ritz discloses [col. 06, lines 14-22] the diagnostics policy service determines when an actual problem has occurred by, for example, detecting a predetermined single error condition, or by detecting a predetermined sequence of error conditions has arisen [exception chain exists]. Ritz further discloses [col. 06, lines 23-29] once a problem is detected, the computing system performs a functional result-oriented step for programmatically diagnosing a problem evidenced by the one or more error conditions.

As per claim 11: **The method of Claim 1, wherein said processing further includes comparing said critical exception data with data contained within an exception information database to determine whether said exception is said critical exception [col. 07, lines 32-63].**

As per claim 12: **The method of Claim 1, wherein said examining further includes labeling said exception as at least one of a critical exception, a non-critical exception, a derived exception and a primary exception.**

Ritz discloses [col. 07, lines 46-63] if the query results are associate with an identified root cause, the invoked diagnostics module may invoke an appropriate resolution module to perform

an identified resolution that corresponds to the identified root cause. The identified root cause for a problem is some problem that is known to exist [derived exception event]. It is understood that if a problem is not known to exist, it would be a primary exception event.

Ritz further discloses [col. 09, lines 6-9] if the error event does not have a known root cause associated with it [critical exception], diagnostics module will report this information to the activity log, which in turn sends an error report to error reporting service. It is understood that if the error even does have a known root cause, it can be fixed with predetermined instructions to follow [non-critical exception].

As per claim 13: **The method of Claim 12, wherein said processing further includes updating said exception information database with said exception data** [col. 08, lines 53-60].

As per claims 14-16 and 18: Although claims 14-16 and 18 are directed towards a system claim, they are rejected under the same rationale as the method claims 1, 3, 2, and 4, respectively.

As per claim 19: Although claim 19 is directed towards a machine-readable medium claim, it is rejected under the same rationale as the method claim 1.

Response to Arguments

5. Applicant's arguments filed on September 14, 2009 have been fully considered but they are not persuasive. In response to applicant's argument that Ritz can't dynamically detect an exception event, the examiner respectfully disagrees. Ritz discloses [col. 06, lines 38-50] the

monitoring service applies policy to filter which events are propagated up to invoke diagnostic module for root cause determination. Examples of such events may be desirable include enterprise environments where an IT manager or system administrator may prefer that the operating system not perform certain automatic root cause determination and/or problem resolutions actions automatically. Therefore, the operating system is capable of dynamically identifying root cause determination.

6. In response to applicant's argument that Ritz does not teach or suggest dynamically identifying application exception and examining said exception data prior to said application exception being logged, the examiner respectfully disagrees. Ritz discloses [col. 05, lines 27-43] the event provider communicate events to a logger. Accordingly, any given event provider need not generate an event for every interaction it senses, but may generate only the more relevant events relating to root causes of problems [identifying critical exception]. For example, an event need not be generated every time a disk drive writes to a sector [non-critical event]. However, an event might be generated if the disk drive fails to respond to a read or write command [critical event]. The logger makes these determinations before the event has been logged to the event trace log file.

7. In response to applicant's argument that in Fig. 3 of Ritz the events are logged immediately after the generation of the events, the examiner respectfully disagrees. Ritz explicitly discloses that only some of the events are logged (not all the event are logged). Therefore, it is understood that the logger determines which events should be logged and which events should not be logged [identifying critical events before logging].

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JIGAR PATEL whose telephone number is (571)270-5067. The examiner can normally be reached on Mon-Fri 10:00AM-6:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Scott Baderman can be reached on 571-272-3644. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Scott T Baderman/
Supervisory Patent Examiner, Art Unit 2114

/Jigar Patel/
Examiner, Art Unit 2114